SIMULATION OF AGENT-BASED CONTROLLED PRODUCTION NETWORKS BY DISTRIBUTED SIMULATION MODELS

Günther Seliger
Guenther.Seliger@iwf-mt.tu-berlin.de
Markus Ciupek
Markus.Ciupek@iwf-mt.tu-berlin.de
Institute for Machine Tools and Factory Management
Technical University of Berlin, GERMANY

The increasing concentration on core competencies in the production industry leads to production networks. The ability to react properly and immediately to changes in demand as well as to technical and organisational disturbances requires a suitable decision support system within the network. In this paper a simulation based decision support environment is introduced. It consists of a High Level Architecture (HLA) application for coupling of simulation tools, in which simulation models of suppliers, transportation agencies and a final producer are integrated as decentralised acting partners. Furthermore an agent based control for coupled simulation models is introduced. The software agents react to changes within the simulated production network. A simplified simulated production network for an assembly of motor-car serves as an example to illustrate this approach.

1. INTRODUCTION

Today it is essential to increase the efficiency of businesses at every opportunity. To meet customer demands, production networks are discussed in industry (Bryne, 1999; Davidow, 1992; Wiendahl, 1996). Production networks are temporary networks that occur to respond to an customer order. This places high demands on the computer network, the Internet or Intranet. Furthermore often happen disturbances such as demand fluctuations in the final production, break down of resources at the supplier side or failures of trucks within the established networks. For these disturbances must be find a fast adjustments to secure a proper work along the value-added-chain.

This paper describes how the Internet can be used to connect commercial simulators and what kind of control for these connected simulators is necessary in order to find adjustments possibilities for the simulated disturbances. Coupling commercial simulators makes it very efficient to simulate customer orders throughout the production network, without additional effort in modeling and simulation. Manufacturers and simulation specialists who are part of the network, bring their simulation capacity into the network. Thus new levels of collaboration can be achieved.

2. PRODUCTION NETWOKRS

2.1 Time Management

Wiendahl defines production networks as a unity of companies which can be dynamically re-organised according to an order (Wiendahl, 1996). Each company concentrates on their core business and, thus, these networks can be seen as a unity of the best.

A central problem in production networks is the coordination. The execution of customer orders in production networks is highly parallel per se. Therefore in a production network, where different simulators are used for modeling different elements of the network it is vital to think carefully about how to synchronize the simulators.

Traditionally discrete event simulation tools use global event lists for keeping track of all scheduled events. All events are executed in a global time stamp order. The simulator also takes care of the repeatability of the results, when events with the same time stamp occur. The same is very important for a distributed version of the simulation model (Fujimoto, 1997).

Different parallel simulation protocols have been developed in the past. They can be classified into conservative and optimistic ones. The conservative ones need to process in time stamp order, whereas the optimistic ones allow out of order processing as long as the resulting state reflects the effect of executing the events in their time stamp order. Otherwise rollback mechanisms have to be applied.

For our experiments we used HLA as the latest development in the area of distributed simulation.

HLA offers mechanisms for all the synchronization problems mentioned above. In our work and the testing we applied a conservative protocol with *lookahead.* This approach was chosen because the simulation tools that were used in our tests do not offer any built-in support for optimistic protocols.

2.2 A Meta-Model in HLA

From the modeling point of view a production network can be seen as a graph consisting of a set of vertices and a set of edges. Each vertex depicts a company and has ability to carry out orders and to place orders to others. Thus the orders are the edges of the graph. So we model two classes. The company class is identified by a name and a set of products. In the order class the attributes *number* and *type* clarify what product should be assembled to what amount. The information and material flow is managed by the attributes *due date* and *delivery date.*

The HLA federation consists of both classes for which each participant has to publish and subscribe. To start a new simulation each simulator creates an instance of the company class. When all the simulators have joined the federation, order objects can be created. To send an order a simulator has to register the new object.

The receiver takes the ownership of the attribute *delivery date*. The sender is notified of the fulfilment of an order by writing the delivery date into that attribute. The order object is destroyed by the sender.

In HLA it is distinguished between remote and local objects. As shown in Figure 1 federates are informed by the federation management about the instances of order and company classes created by other federates. These objects are remote objects. As a logical consequence each federate holds all objects of the federation. It is the task of the federation management to keep the federates informed about ongoing changes in existence and state of the objects.

In our case the model consists of three company objects and one order object. The order is placed by Simulator A and therefore this object is a local one. The arrow to the federation management indicates that changes in the model are sent to the federation management by the view object the *RTIambassador*. The arrow to the controller objects the *FederateAmbassador* shows that the federation management passes all changes to each federate.

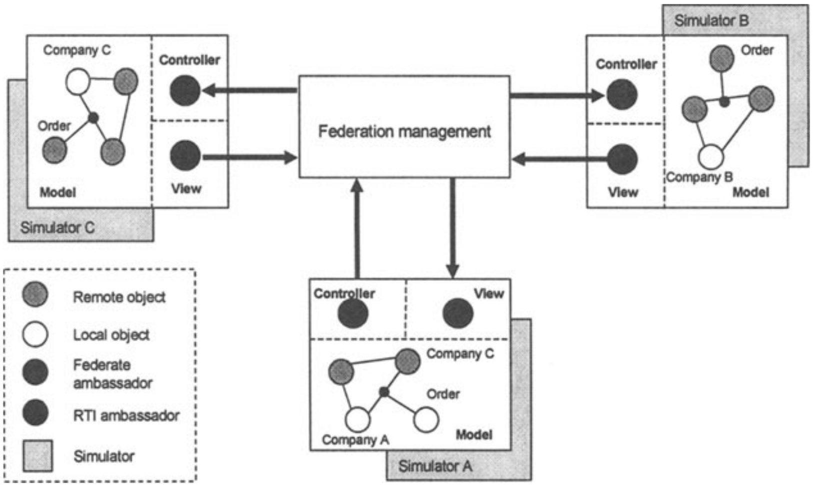


Figure 1 – Realization of the meta-model in HLA

3. HLA INTERFACE OF SIMULATION TOOLS

3.1 General Approach

HLA defines a two-part interface which federates are required to use for communicating with the Runtime Infrastructure (RTI).

The HLA interface is based on an ambassador paradigm. A federate communicates with the RTI using its RTI ambassador. Conversely, the RTI communicates with a federate via the federate's ambassador. From the federate programmer's point of view these ambassadors are objects and the communication between the participants is performed by calling methods of these objects.

In order for commercial simulation tools to fit into this ambassador paradigm it is a good approach to use wrapper libraries (Straßburger, 1998). The wrapper library has the task to translate between the RTI functionality and the capabilities of the simulation tool.

While the wrapper library can take care of the software technological connection to the RTI, it is also necessary to consider the problems that result from the simulation side. The next section deals with the problems resulting from being member of a distributed simulation and gives a classification of simulation tools regarding their capabilities to process external events.

3.2 Open Event Loop vs. Closed Event Loop

Participants in a distributed simulation have to coordinate their local advances in logical simulation time. To do so they have to take into account their mutual dependencies.

The HLA approach includes a transparent time management that allows federates to coordinate their logical simulation clocks. In order for this to work, the RTI requires federates to request their time advances by calling the appropriate methods of the RTI ambassador object. Issuing such a request may result in

- a grant of the request
- a notification about an external event, that needs to be processed prior to the local event that the request was issued for.

This is the place where our research shows that it is necessary to distinguish between two types of simulators:

- Open event loop simulators
- Closed event loop simulators

Open event loop simulators allow the user to integrate the external event in the local events chain. The external event can then be processed. Afterwards the simulation tool can proceed with the next local event.

An example for an open event loop simulator is SLX (Henriksen, 1996). Our solution for SLX is very straight-forward: The user issues a *nextEventRequest* for a local event. If an external event needs to be processed, the return value of the function is the time stamp of this event. If no external events where received, the return value equals the time stamp of the request. In the former case, the simulator is told to advance to the time stamp of the external event and to process it. In the latter case the simulator can proceed without interruption to the time stamp of the local event.

Closed event loop simulators do not allow the integration of external events into the events chain of the simulator. They may not even have a built-in function to determine the time stamp of the next local event. Examples for such tools are AutoMod and ProModel.

For this type of simulator we suggest a different approach for synchronization. Since we are not able to determine the time of the next local event we use the *nextEventRequest* with a virtual next event time. Synchronization occurs in intervals of T time units. All outgoing events (like sending orders or requests) for the time T+t are bundled before their release. Thus the parameter T directly determines the bundling size.

4. DESIGN OF AGENT-BASED CONTROL

4.1 General Approach

Wiendahl and Ahrens denoted an agent as a self-contained computer program that can solve certain tasks independently and passes the solutions to other agents, either by its own initiative or when requested by other agents (Wiendahl, 1997).

In (Teti, 1997) the definition of an agent is applied to manufacturing systems. Agents are given responsibility of scheduling and controlling a manufacturing system. Other agents fulfil certain subtasks. Yet other agents are responsible for recording, keeping accounts and reporting on the system.

Negotiations are regulated by protocols. A complete agent environment is described in (Wiendahl, 1997). It is based on message passing techniques. The main classes in this environment are agents and conversations. For each conversation, agents act according to conversation guidelines. The conversation guidelines are described by states and transitions between states.

4.2 Agent-based Control

Agents are self-contained computer programs that can solve certain tasks independently and passes the solutions to other agents, either by its own initiative or when requested by other agents. An agent environment is based on message passing techniques. The main classes in this environment are agents and conversations (Barbuceanu, 1996). Conversations between agents are regulated by protocols. For each conversation, agents act according to conversation guidelines. The conversation guidelines are described by states and transitions between states.

The developed agent society for the assembly control is shown in Figure 2. Three levels are distinguishable.

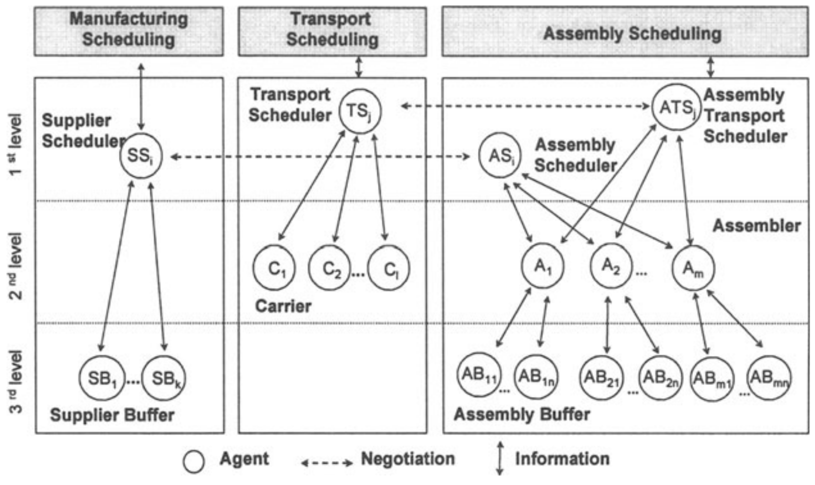


Figure 2– Society of agents for assembly control

At the first level are the negotiation agents: *Assembly Scheduler-Agent* (AS), *Assembly Transportation Scheduler-*Agent (ATS*), Transportation Scheduler-Agent* (TS) and the *Supplier Scheduler-Agent* (SS). The second level belongs to the *Carrier-Agents* (C) and the *Assembly-Agents* (A). The latter tries to ensure that there are the right parts in assembly. The Carrier-Agent supervises the complete tour of delivery. These agents are called controller agents. On the third level are the stock agents represented by the *Supplier Buffer-Agents* (SB) and the *Assembly Buffer-Agents* (AB).

These stock agents pass information on the actual inventory levels to agents requesting them. The controller and stock agents pass the necessary information about their current status to the negotiation agents. All mentioned agents act according to conversation guidelines represented by a set of rules. Only the negotiation agents are able and authorised to make decisions e.g. the TS-Agent allocate transportation batches to trucks.

4.3 Structure of the Agents

In order to achieve a dynamic adjustment to modified conditions, the agent control system must be able to evaluate a situation fast enough to undertake the right decisions.

The structure of the developed software agents is composed of two layers: the communication layer and the behaviour layer. The communication layer handles communication between other agents. The second layer implements the reactive behaviour of the agent by calling methods at specific time or when messages are received. The behaviour of the agents is mainly designed by fuzzy logical rules. These rules are conditional statements of the form IF A THEN B, where A and B have fuzzy meaning e.g. IF x is small THEN y is large (Zadeh, 1973). Every agent has its own set of rules in dependence on the fulfilling task.

The above introduced controller and stock agents use the behaviour layer for the evaluation of its own situation. The evaluation process is carried out in three steps: transformation of numerical values into fuzzy values, parallel execution of all fuzzy conditional statements (fuzzy inferences) and retransformation of the fuzzy inferences into a numerical value. At the end of the evaluation process every controller agent or stock agent pass its specific numerical value (further called negotiation value) to its corresponding negotiation agent.

Figure 3 show the evaluation of the negotiations value of an carrier agent. In the first step the current numerical value of the remaining transportation capacity of a truck e.g. 13 m^3 will be transformed by executing all four membership functions to the fuzzy value "good". During the second step the rules of inference are applied.

For the described example only rule 3 is executed and assigns the fuzzy statement "normal". In the last step the statement "normal" is transformed back by executing to this statement belonging membership function. The numerical result of 3 is current negotiation value of the carrier agent.

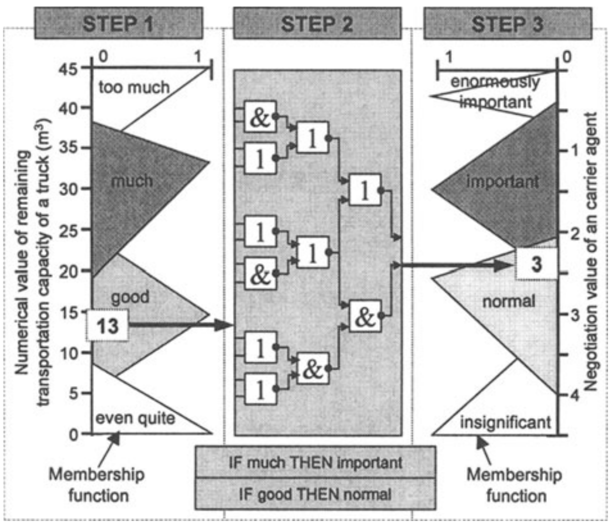
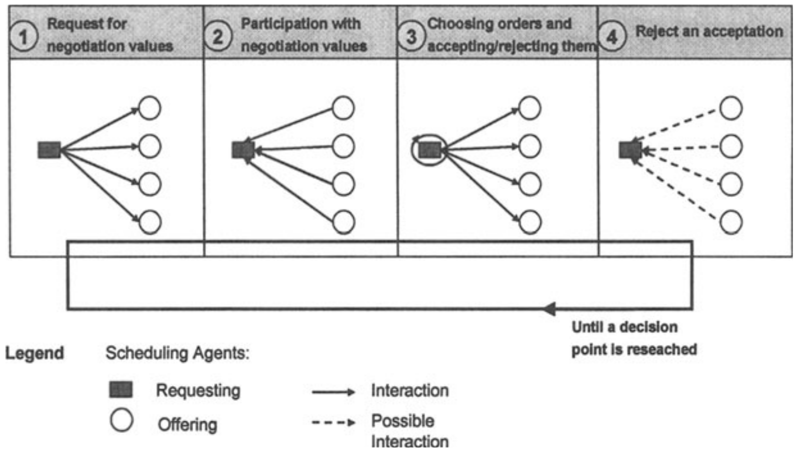


Figure 3– Determination of a negotiation value

4.4 Negotiation

The aim of the negotiation is to find the best allocation of jobs, like transportation jobs in accordance to resources e.g. number of trucks according to orders from the assembly scheduling. This negotiation will be initialised by the controller agents depending on the current situation of the transportation agency, suppliers and final assembler. For example if the availability of trucks, the suppliers supplying dates or the final assemblers planned schedule has changed.

The Negotiations between the negotiation agents (*Assembly Scheduler-Agent*, *Assembly Transportation Scheduler-*Agent, *Transportation Scheduler-Agent* and the *Supplier Scheduler-Agent)* are executed in four phases (see Figure 4).



1 Request for negotiation values
2 Participation with negotiation values
3 Choosing orders and accepting/rejecting them
4 Reject an acceptation
Until a decision point is reseached
Legend
Scheduling Agents:
Requesting
Offering
Interaction
Possible Interaction

Figure 4 – Negotiation pattern

In the first phase the negotiation agent which was informed about a change by its controller agents sends a request to other negotiation agents. These agents asks their controller and stock agents for their current negotiation values. In the further phases the requesting negotiation agent makes its selection on the basis of these values and informs the other negotiation agents about refusal or acceptance of their offers. The negotiation can last until defined decision points are reached. This is important to finish a negotiation and to avoid a dead lock of the control system. These decision points serve to determine certain resources - job combinations which cannot be resolved any more.

5. TESTING THE SIMULATION ENVIROMENT

The simplified final assembly of motor-cars serves as an example for the testing of the developed simulation environment. Therefore have been developed three simulations models i.e. a supplier model, a transportation agency model and an assembly model. The supplier model represents a supplier with five product families, *viz.* car body, tire, radiator, brake and wheel case. The assembly model depicts the final assembly of the motor-cars. The model of the transportation agency is responsible for delivering the orders between the supplier and the final assembly. These three simulation models are coupled together with HLA by using an configuration of the following three federates.

- Federate A for the supplier
- Federate B for the transportation agency
- Federate C for the final assembly

The challenge in this example is to schedule the trucks of the transportation agency model and to decide which batches to transport under occurring disturbances in all three simulation models.

The following three situations are compared in order to test on one hand the HLA-based connection of the simulation model and on the other hand the developed agent control:

1. The transportation agency model receives a fixed transportation plan, in which charges are assigned to trucks. The modeled network between supplier, transportation agency and final assembly is simulated without any disturbances.
2. The transportation agency model receives a fixed transportation plan. The transportation plan is strictly fulfilled without any adaptation to occurring disturbances.
3. The developed agent based control is initialized by the transportation plan. The developed agents try to adapt this plan to the disturbances by applying the negotiation pattern. The given plan will be resolved and adapted dynamically to the situation.

During all undertaken simulation tests the HLA-based coupling of the simulations models worked without any interruption. The performance measurements of the agent based control has been carried out with the objective intime delivery. With this objective the percentage of intime delivered orders for the final assembly was measured. In the first two runs of the model the first and second above assumed situation was simulated. The rest of the runs were regarded to test the performance of the agent control. Therefore the use of the variables stocks in the assembly (A), stocks by the suppliers (S) as well as the utilization of the trucks (T) was tested. For every variable its own fuzzy logical rules was implemented.

Figure 5 shows the evaluation of the results of the simulation. Six simulation runs were executed. The best result as supposed has been achieved during the simulation run with fixed transportation plan and no disturbance i.e. 100% of intime delivery, and the worst result occurred during the simulation with fixed transportation plan and disturbances. Furthermore Figure 5 shows that every simulation run using the agent control performed much better as compared to the result shown by fixed transportation plan with disturbance. It is also remarkable that the parallel application of all rules (S+A+T) of the agent based control led to the best result.

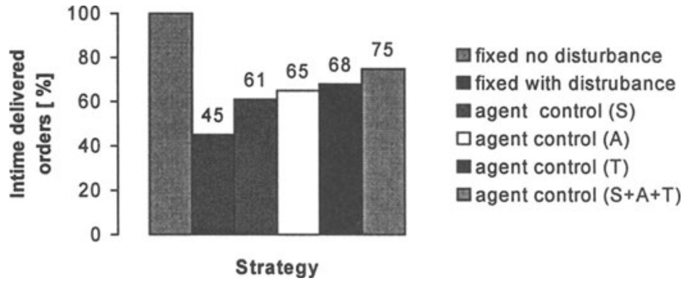


Figure 5 – Simulation results

Figure 6 shows the result for the duration of 25 shifts. The inventory statistics of the wheels serves as an example. Curve X_a shows the case without agent-based control. The curve X_b depicts the case with agent-based control. It is obvious that we can operate the system with lower stocks by the agent base control. Through agent-based control capacity reserves can be managed efficiently. In this case, the system is also able to adjust the use of trucks capacities.

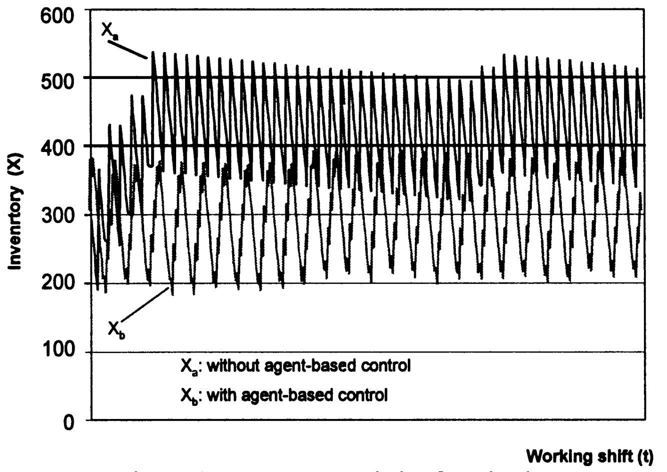


Figure 6 – Inventory statistics for wheels

5. CONCLUSION AND OUTLOOK

HLA offers a new simulation interoperability standard. Simulation runs have proved the possibility of connecting commercial simulators with HLA. Furthermore has been introduced an agent-based control for coupled simulation models in order to solve unexpected problems which occur dynamically during the supply chain, such as fluctuations in demand, supply and process uncertainties. By using the fast reacting agent control it has been realized an average reduction of the inventory of 30%

Future competitiveness of enterprises will mainly be determined by how fast they are able to interlink their core competencies. Therefore in further research will be examined to what extent an agent based approach can be helpful in establishing virtual enterprises. Such a agent based approach will comprise both logistical and product developing networks.

6. REFERENCES

1. Barbuceanu, M., Fox, S. M.: Capturing and Modeling Coordination Knowledge for Multi-Agent Systems. In: International Journal of Cooperative Information Systems, Vol.5 (1996), No.2-3 , pp. 273-314.
2. Byrne, J. A., Brand, R., Port, O.: The Virtual Corporation. In: Business Week, (1999) 5, p. 37.
3. Davidow, W. H., Malone, M. S.:The Virtual Corperation: Structuring and revitalizing the corporation for the 21st century, New York, 1992.
4. Fujimoto, R. M.: Zero Lookahead and Repeatability in the High Level Architecture. In: Proceedings of the Spring 1997 Simulation Interoperability Workshop, Orlando 1997, Paper No. SIW97S-046, available online at http://www.dmso.mil/projects/hla/papers/.
5. Henriksen, J.O.: An Introduction to SLX. In: Proceedings of the 1996 Winter Simulation Conference, New Jersey 1996, pp. 468-475.
6. Straßburger, S., T. Schulze, U. Klein, J.O. Henriksen: Internet-based Simulation using off-the-shelf Simulation Tools and HLA. In: Proceedings of the 1998 Winter Simulation Conference, Washington D.C. 1998, pp. 254-267.
7. Teti, R., Kumara, S. R. T.:Intelligent Computing Methods for Manufacturing Systems. In: Annals of the CIRP, Vol. 46/2 (1997), pp. 629-652.
8. Wiendahl, H.-P., Ahrens, V.: Agent-based Control of Self-Organized Production Systems. In: Annals of the CIRP, Vol. 46/1 (1997), pp. 365 - 368.
9. Wiendahl, H.-P., Kuhn, A., Fastabend, H., Helm, K., Kloth, M: Kooperatives Management in wandelbaren Produktionsnetzen. In: Industrie Management, (1996) 6, pp. 23-28.
10. Zadeh, L. A.: Outline of a New Approach to the Analysis of Complex Systems an Decision Prosses. In: IEEE Transaction on Systems, Man, and Cybernatics, Vol. SMC-3 (1973), No.1, pp. 28 - 43.